

OSDcaptcha: Combination of Orientation, Sequence & Drag Drop and its Analysis

Ashwini Kumar¹, Arpit²

Department of Computer Science

Institute of Engineering & Rural Technology (IERT) Allahabad (U.P.), India

¹simplyashwini@gmail.com, ²arpittabelabux@gmail.com

Abstract-We discuss a new architecture CAPTCHAs based on orientation of subject of image's object. Existing CAPTCHAs that based on image, suffer from two limitations, the first one is the availability of less public image existence and second one is that wrong label /caption of images. Miss labeled can become the culprit for Human frustration. To here is need of other type of image CAPTCHAs called OSDcaptcha over these limitations. OSDcaptcha is combination of Orientation, Sequence and Drag Drop become practical for new generation of system and very handy for touch screen devices. Our proposed CAPTCHAs is based on orientation of subject of image's object. A finite slicing of image into sub image can be per mutated and present in front of user in infinite possibility. The random sequencing of sub image in Drag Drop approach leads to distinguished between pernicious bots and Humans. Automated attack must have to recognize all sub object present in sub image in sequence, reconnect via sequencing and lastly drop the sub images according to overall final image content's orientation, which is difficult to bots or nefarious program to crack three tier securities.

Keywords-Drag Drop; Security; CAPTCHA; HIP; OCR; Sequence

I INTRODUCTION

Human Interaction Proofs (HIPs) are the methods that allow a node /system to distinguished user is a Human not a automated program or bots. Services may require a user to pass a CAPTCHA [1] (Computer Automated Turing Test for telling Computers and Humans Apart), a challenge-response test that can be easily pass by human users but is tough for automated bots. Using CAPTCHAs, we can resolve many type of security breach for example auto registration, brute force password attacks and abuse of online polls. In order to be effective in this role, it is important that CAPTCHAs are strongly resistant to automated attacks and at the same time do not cause problems for human users.

In September 1999, www.slashdot.com conducted a survey to figure out in which university graduates of the department of computer science were the best. Although the voting system could prevent the same IP address from voting more than once, students from CMU wrote a program to make the number of votes for CMU go up rapidly. The next day, students from MIT adopted a similar approach, resulting in that the number of votes for either of these two universities far exceeded the other universities. Moreover, a report from the Barracuda Network Security Corporation in the USA said that in 2007 nearly 95% of the mails received by the world's

Internet users were junk mails. Similar situations are registering user accounts maliciously, cracking account passwords with brute force, etc. All of these bring a great threat to the network.

In order to prevent similar incidents from happening again, CAPTCHA mechanism comes into being, which is short for Completely Automated Public Turing Test to Tell Computers and Humans Apart [2, 3]. In 2000 Carnegie Mellon University set up the first CAPTCHA group, followed by many scholars studying CAPTCHA to find how to better tell between humans and computers apart.

Broadly, CAPTCHAs can be classified into two groups - OCR-based CAPTCHAs and Non-OCR based CAPTCHAs [5].

Text-based CAPTCHAs, in which users are vital to transliterate text obtainable in a distorted image, make up the majority of CAPTCHA systems in real-world use. However, their vulnerability to attack has been repeatedly demonstrated by computer vision researchers [2, 3, and 4]. To safeguard against automated attacks, tougher and more sophisticated distortions has been employed in creating images used by text-based CAPTCHAs; this, however, greatly impairs human legibility, resulting in higher user error rates [5] and associated levels of user frustration. Further, for mobile phones and devices like PDA's and palmtops, the use of keyboard is infeasible thus making OCR-based CAPTCHAs inconvenient. These weaknesses can be resolved by using Non-OCR based CAPTCHAs.

These include image CAPTCHAs that exploit three facts:

- The human eye is naturally very good in recognizing object present in image.
- The hefty variety of image present in public domain
- (a) A finite slicing of image can be per mutate and present in front of user in infinite possibility.
- (b) The random sequencing of sub image in Drag Drop approach leads to distinguished between pernicious bots and Humans.

II RELATED WORKED AND PROBLEM

Many early image-based CAPTCHA systems simply ask the user to identify the subject of an image or a subject associated with a set of images. However, resistance to image-similarity based attacks requires the creation and maintenance

of a large, correctly labelled image database – a task that is labour intensive and fraught with legal issues regarding usage. Approaches to dealing with this shortcoming include the sourcing of images from image search engines [6]; however, the manner in which image search engines tag images may result in mislabelled or potentially offensive results.

Microsoft's Asirra [7] asks users to identify the images of cats out of a set of photographs of cats and dogs, sourced dynamically from a frequently updated pet adoption website. However, a successful machine learning attack [8] has been demonstrated against the system, a success that largely appears to be a result of the binary nature of the classification problem posed. The composition-based nature of our system largely avoids image database creation problems by generating an extremely large space of possible test images and associated questions from a relatively small database size.

Google Research's What's Up CAPTCHA [9] requires users to identify an image's proper upright orientation. One disadvantage of the system is that attackers may be able to build a pre-rotated image database given a known image database and then utilize image similarity metrics in order to determine correct image rotations. The task's user success rate when the images have undergone distortion, the proposed defence, has not been tested. It is possible that usability would become a problem in this case. However, if the image database is publicly available, the database can be algorithmically searched resulting in the automated classification of the image. In addition, huge image databases, such as images.google.com, often generate incorrect labels for any given image.

Although text based CAPTCHA are currently most widely used ones, but the advances in OCR techniques in terms of pattern recognition and computer vision have made them prone to more and more attacks [12, 13, and 14].

Currently, CAPTCHA creators recommend use of reCAPTCHA as the official implementation [15]. In September 2009, Google acquired reCAPTCHA to aid their book digitization efforts [16]. However, this CAPTCHA has been cracked with 30% success rate, reported in August 2010 [17].

Currently, there are mainly three kinds of methods to implement the CAPTCHA mechanism: OCR (Optical character recognition) visual method, non-OCR visual method and non-visual method.

The 2D static CAPTCHA based on OCR visual method takes advantage of superiority in language barrier, security and easy use, becoming the most widely used CAPTCHA [4], as shown in Fig. 1 (a). Commonly seen CAPTCHAs are: Gimp series CAPTCHA designed by Carnegie Mellon University in 2000, Pessimal Print CAPTCHA designed by Henry Baird from PARC (Palo Alto Research Center) in 2000, and Baffle Text CAPTCHA designed by Baird in cooperation with Monica Chew from California Berkeley in 2003 [5]. However, with the fast development of OCR technology based on neural network, as well as the emergence of a variety

of character segmentation technology, CAPTCHAs of lots of websites have been attacked. A Russian programmer has ever cracked the CAPTCHA mechanism of Yahoo with 35% success rate. Also, the CAPTCHA mechanism of Microsoft live mail has been bothered by junk mails many times. Given facts like these, newly designed CAPTCHAs have become increasingly complex, so that some of those are extremely difficult to identify.

Though there are many different kinds of specific implementations for non-OCR visual method, it eventually comes down to the OCR problem in general, requiring users to identify images. It is not so widely used. Up to now, except some research sites, commercial sites rarely use it. Specific implementation algorithms are: CAPTCHA algorithm based on real object image identification and designed by R. Datta, etc, CAPTCHA algorithm based on image similarity judgment and designed by J. Elson, etc, and so forth. Non-OCR visual method is designed for special occasions and certain user groups, thus it has very limited applications. Examples are: voice-based CAPTCHA algorithm intended for visually disabled people and designed by G. Kochanski, etc, CAPTCHA algorithm based on collaborative filtering and designed by M. Chew, etc, and so forth.

In conclusion, the OCR-based 2D static visual method is the main way to implement current CAPTCHA mechanism. However, it could no longer strike a balance between security and easy use, calling for a new kind of CAPTCHA to address this increasingly prominent problem.

III APPLICATIONS

CAPTCHAs are used in various Web applications to identify human users and to restrict access to them. Some of them are:

Online Polls: As mentioned before, bots can wreak havoc to any unprotected online poll. They might create a large number of votes which would then falsely represent the poll winner in spotlight. This also results in decreased faith in these polls. CAPTCHAs can be used in websites that have embedded polls to protect them from being accessed by bots, and hence bring up the reliability of the polls.

Protecting Web Registration- Several companies offer free email and other services. Until recently, these service providers suffered from a serious problem – bots. These bots would take advantage of the service and would sign up for a large number of accounts. This often created problems in account management and also increased the burden on their servers. CAPTCHAs can effectively be used to filter out the bots and ensure that only human users are allowed to create accounts.

Preventing comment spam: Most bloggers are familiar with programs that submit large number of automated posts that are done with the intention of increasing the search engine ranks of that site. CAPTCHAs can be used before a post is submitted to ensure that only human users can create posts. A CAPTCHA won't stop someone who is determined to

post a rude message or harass an administrator, but it will help prevent bots from posting messages automatically.

Search engine bots: It is sometimes desirable to keep web pages unindexed to prevent others from finding them easily. There is an html tag to prevent search engine bots from reading web pages. The tag, however, doesn't guarantee that bots won't read a web page; it only serves to say "no bots, please." Search engine bots, since they usually belong to large companies, respect web pages that don't want to allow them in. However, in order to truly guarantee that bots won't enter a web site, CAPTCHAs are needed.

E-Ticketing: Ticket brokers like TicketMaster also use CAPTCHA applications. These applications help prevent ticket scalpers from bombarding the service with massive ticket purchases for big events. Without some sort of filter, it's possible for a scalper to use a bot to place hundreds or thousands of ticket orders in a matter of seconds. Legitimate customers become victims as events sell out minutes after tickets become available. Scalpers then try to sell the tickets above face value. While

CAPTCHA applications don't prevent scalping; they do make it more difficult to scalp tickets on a large scale.

Email spam: CAPTCHAs also present a plausible solution to the problem of spam emails. All we have to do is to use a CAPTCHA challenge to verify that indeed a human has sent the email.

Preventing Dictionary Attacks- CAPTCHAs can also be used to prevent dictionary attacks in password systems. The idea is simple: prevent a computer from being able to iterate through the entire space of passwords by requiring it to solve a CAPTCHA after a certain number of unsuccessful logins. This is better than the classic approach of locking an account after a sequence of unsuccessful logins, since doing so allows an attacker to lock accounts at will.

As a tool to verify digitized books-This is a way of increasing the value of CAPTCHA as an application. An application called re-CAPTCHA harnesses users responses in CAPTCHA fields to verify the contents of a scanned piece of paper. Because computers aren't always able to identify words from a digital scan, humans have to verify what a printed page says. Then it's possible for search engines to search and index the contents of a scanned document. This is how it works: The application already recognizes one of the words. If the visitor types that word into a field correctly, the application assumes the second word the user types is also correct. That second word goes into a pool of words that the application will present to other users. As each user types in a word, the application compares the word to the original answer. Eventually, the application receives enough responses to verify the word with a high degree of certainty. That word can then go into the verified pool.

Improve Artificial Intelligence (AI) technology: Luis von Ahn of Carnegie Mellon University is one of the inventors of CAPTCHA. In a 2006 lecture, von Ahn talked about the relationship between things like CAPTCHA and the field of

artificial intelligence (AI). Because CAPTCHA is a barrier between spammers or hackers and their goal, these people have dedicated time and energy toward breaking CAPTCHAs. Their successes mean that machines are getting more sophisticated. Every time someone figures out how to teach a machine to defeat a CAPTCHA, we move one step closer to artificial intelligence.

As people find new ways to get around CAPTCHA, computer scientists like von Ahn develop CAPTCHAs that address other challenges in the field of AI. A step backward for CAPTCHA is still a step forward for AI: "every defeat is also a victory".

IV PROPOSED METHOD

In this section, we present a new CAPTCHA technique that aims to determine legitimate users and at the same time do not alienate them in the proposed technique. A subject-wise database of small sized, well known small real world objects is created. The images and the corresponding tags (object) associated with each sub image are stored in separate places which are related with each other on an encoded key within a single database so that even if a hacker is able to break into the database, he is not able to get any meaningful content. An as option a dynamic image database could be created by downloading images from the Internet [10, 11]. In the proposed technique a composite CAPTCHA image of a reasonable dimension and resolution is shown to the user.

A. Challenge Creation at Server Side

- 1) Select a reasonable dimension and resolution image from public domain, rename it and store in dataset.
- 2) Horizontally slice in three or four sub image store it along its own dataset sequence and per mutate it.
- 3) Apply few standard images processing mechanism to distort the sub images.
- 4) Set it for Drag Drop mechanisms.

B. Challenge Solve at Client Side

- 1) User firstly identifies the object from the sub images.
- 2) Now users determine appropriate orientation of object.
- 3) In final step user must have to Drag and Drop the all image according content orientated top to bottom sequence to clear the test.
- 4) Always concern about session expiration to avoid pornography attack.

C. Removing Confused Images

- 1) For solving complication there is need of social condemn system (optional for users).
- 2) Images that have a high variation in their content orientations are those that are likely to have no clear upright orientation, can identify and exclude difficult images from our dataset using above condemn system.

D. Example: OSDcaptcha Based on Baby Image



Figure 1 Sub images and bogus sub images in Random sequence [(a) (b)],[(c),(d)],[(e),(f)]



Figure 2 Sequence (a), (f) and (e) to form complete image according to top to bottom image's content

V SECURITY ANALYSIS

Security of a CAPTCHA technique can be analysed in terms of time taken, resources involved and efficiency of a auto bots program that tries to breach the test. Any CAPTCHA technique is considered to be securing that is at least as expensive for a hacker as it would cost him using human operators. In this section we will present the security analysis performed on the proposed CAPTCHA technique. If the hacker employs random guessing to solve the CAPTCHA, then since an object will be disassemble into parts in the CAPTCHA round, he/she will choose to click a minimum of three sub image (corresponding to the case where object form) and a maximum of all sub image to get the object . This is because an object is disassembling into at least three sub

images. In OSDcaptcha needs sufficient size of image 10*6 cm in dimension .Means there will possible of slicing 10*10= 100 points selecting 3 point random point a one side, 6 for both side. If challenge sub images count equal to 'n', 't' sub images require for assembling, and various transformation 'd1' & 'd2' degree of stress, 's1' & 's2' degree of scaling and final 'r' of noise addition . For limited standard value 10 taken for each case:

$$\begin{aligned}d1 &= 10; \\d2 &= 10; \\s1 &= 10; \\s2 &= 10; \\r &= 10;\end{aligned}$$

So total combination

(c) = $d1*d2*s1*s2*r = 1,00,000$ case generates form various transformation. Since there will be one correct sequence, the random guessing probability Number of possibility. To guess correct transformation:

$$x = \frac{1}{\sum_{t=3}^n \binom{n}{t} t!}$$

Capturing correct Slicing probability is:

$$\frac{1}{100000}$$

Hence, to solve the CAPTCHA, 't' sub images are required to be clicked in a particular sequence:

$$\frac{3}{100}$$

Now final Probability (p) of slicing *probability of transformation *probability of corresponding correct sequence:

$$p = \frac{1}{\sum_{t=3}^n \binom{n}{t} t!} * \frac{1}{100000} * \frac{3}{100}$$

For sample n=6 and t=3 as above condition which approximately equal Three million to one condition? If session expires within 30 sec, the numbers of calculation require is hundred thousand per sec, which hardly to determine by bots using any attack using Object Recognition before session expires. The Pictionary-based attack is not feasible to break this scheme because the dissection points of object pictures are dynamic and chosen at run-time. Hence, even if the same image is selected to be displayed, the slicing point is chosen randomly, random stress, scaling and noise addition make tough and unfeasible for automatically attacks.

VI ADVANTAGES OF OSDCAPTCHA

Unlike text CAPTCHAs, its language is independent so no bother about in which language website shows. OSDcaptcha can integrate easily with on-line systems and websites. OCR-based CAPTCHAs, OSDcaptcha s do not require the user to type anything. This eliminates the need of keyboard. Therefore, OSDcaptcha can solve on hand-held devices or devices in which it is unwieldy to use the keyboard, such mobile, playbook and e-pad. OSDcaptcha does not require any processing on client side. Thus, it is feasible and suitable to use on small devices and devices with limited resources such as mobile etc. By morphing, the pictures can be

sufficiently modified in terms of their color scheme, intensity and introduction of noise and other random shapes and objects such that they convey the same meaning to the user but a bot would not be able to recognize it as an old picture and would consider it as new.

VII CONCLUSIONS

OCR-based CAPTCHAs have been broken and remain insecure. Non-OCR based CAPTCHAs retain convenience of operation for humans as they exploit the natural skill of the human eye of identifying pictures. We employed the concept of Sequencing in Picture CAPTCHAs to introduce OSDcaptcha. CAPTCHAs incorporate three levels of security, viz. recognition of objects in pictures and determining their logical sequence and Drag Drop it. Few more technique in arbitrary image processing creates a new image for bots. Hence, OSDcaptcha are the new area of research and development to increase more security and reduce auto triggered attacks. There are a lot of scope to improve the algorithm random transformation which recognize by human not by auto bots. Our OSDcaptcha still suffer from pornography attack so there is need of strong surrogates scripting to reduce these types of attack.

REFERENCES

- [1] L. von Ahn, M. Blum, and J. Langford. Telling Humans and Computers Apart (Automatically) or How Lazy Cryptographers do AI. *Comm. of the ACM*, 47 (2), 57-60.
- [2] Chellapilla, K., and Simard, P.Y. Using Machine Learning to Break Visual Human Interaction Proofs (HIPs). *Advances in Neural Information Processing Systems 17*, 265-272.
- [3] Moy, G., Jones, N., Harkless, C., and Potter, R. Distortion Estimation Techniques in Solving Visual CAPTCHAs. In *IEEE CVPR*, (Washington, D.C., USA, 2004), Vol. 2, 23-28.
- [4] Yan, J. and El Ahmad, A. S. 2008. A low-cost attack on a Microsoft CAPTCHAs. In *Proceedings of the 15th ACM Conference on Computer and Communications Security* (Alexandria, Virginia, USA, 2008), 543-554.
- [5] Yan, J. and El Ahmad, A. S. 2008. Usability of CAPTCHAs or usability issues in CAPTCHA design. In *Proceedings of the 4th Symposium on Usable Privacy and Security* (Pittsburgh, Pennsylvania, 2008), Vol. 337, 44-52.
- [6] M. Chew and J. D. Tygar. Image Recognition CAPTCHAs. In *Proceedings of the 7th Annual Information Security Conference* (Palo Alto, CA, USA, 2004), 268-279.
- [7] Elson, J., Douceur, J.R., Howell, J., and Saul, J. Asirra: a CAPTCHA that exploits interest-aligned manual image categorization. In *Proceedings of the 14th ACM Conference on Computer and Communications Security* (Alexandria, Virginia, USA, 2007), 366-374.
- [8] Golle, P. 2008. Machine learning attacks against the Asirra CAPTCHA. In *Proceedings of the 15th ACM Conference on Computer and Communications Security* (Alexandria, Virginia, USA, 2008), 535-542.
- [9] Gossweiler, R., Kamvar, M., and Baluja, S. 2009. What's up CAPTCHA?: a CAPTCHA based on image orientation. In *Proceedings of the 18th international Conference on World Wide Web* (Madrid, Spain, 2009), 841-850.
- [10] K. Yanai, M. Shindo, and K. Noshita, A fast image gathering system from the World-Wide Web using a PC cluster, *Image and Vision Computing*, vol. 22, Issue 1, pp. 59-71 January 2004.
- [11] M. Chew, J. D. Tygar, Image recognition CAPTCHAs, in: *Proc. of the 7th International Information Security Conference (ISC 2004)* Springer, 2004.
- [12] G. Moy, N. Jones, C. Harkless, R. Potter, Distortion estimation techniques in solving visual captchas, in: *CVPR (2)*, 2004.
- [13] K. Chellapilla, P. Y. Simard, Using Machine Learning to Break Visual Human Interaction Proofs (HIPs), MIT Press, Cambridge, MA, pp. 265-272, 2005.
- [14] K. Chellapilla, P. Simard, M. Czerwinski, Computers beat humans at single character recognition in reading-based human interaction proofs (hips), in: *In Proceedings of the Second Conference on Email and Anti-Spam (CEAS)*, Palo Alto, CA, 2005.
- [15] CAPTCHA homepage. <http://www.Captcha.net>. Retrieved 2009-12-04.
- [16] Teaching computers to read: Google acquires reCAPTCHA". 2009. Retrieved 2009-09-16.
- [17] Homegrown algorithms for cheating Google's reCAPTCHA: <http://www.darkreading.com>. Retrieved 2010-08-08.